International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 6 Issue: 3 208 - 213

# Survey of Homomorphic schemes

KavitaAganya
Computer Science & Engineering
R.N.Modi Engineering College
Ranpur ,Kota
*kavitaaganya@gmail.com*

Iti Sharma
Computer Science & Engineering
R.N.Modi Engineering college
Ranpur,kota
*itisharma.uce@gmail.com*

***Abstract*—**Homomorphic encryption is increasingly becoming popular among researchers due to its future promises.Homomorphic encryption is a solution that allows a third party to process data in encrypted form. The decryption keys need not be shared.This paper summarizes the concept of homomorphic encryption and the work has been done in this field.
***Keywords*—***homomorphic encryption; security; ciphertext; polynomials*

*****

## I. INTRODUCTION

With established technologies like SSL, Digital signatures conventional cryptography has successfully ensured data privacy and other security services to Internet based applications. Yet, the technical evolution in these applications demands stronger security. Nowadays, smaller enterprises that have limited resources can extend their capabilities temporarily through leasing. Such pay-per-use facility is made possible through techniques collectively called "cloud computing".

The term and techniques are much in use today and newer security challenges get associated day-by-day. The security notions are changing and developing. The data that resides in cloud has multiple ownership and the access for use and permission for processing needs to be granted as a part of security service. Thus, today we need such security methods that can allow processing of data by some third party applications while maintaining its privacy.

Homomorphic encryption is a type of encryption that allows computations to be done on ciphertext directly without decrypting and generate an encrypted result which matches the result of operations performed on the plaintext when it is decrypted.

Homomorphic encryption let us connect together and use different services on a data without exposing data to each of these services or functionalities. For ex: many services from different companies could calculate tax, shipping, currency exchange rate etc.

on a transaction without decrypting the encrypted data to the services so as to maintain security of data.

A homomorphic encryption maintains the security and privacy of data while giving the flexibility of performing computation on data. It can be done by an untrusted third part without revealing its exact input.

## II. BACKGROUND

Homomorphic encryption is a type of encryption that allows computations to be done on ciphertext directly without decrypting and generate an encrypted result which matches the result of operations performed on the plaintext when it is decrypted.

In other words homomorphic encryptions allows us to do homomorphic operations like addition, multiplication etc. without decrypting the ciphertext. And when we decrypt it into plain text it gives same results when we perform those operations on plain text.

A homomorphic encryption system $\varepsilon$ is a combination of four algorithms:

$KeyGen_\varepsilon$: Generates key for the scheme ε.

$Encrypt_\varepsilon$ : Encryption algorithm for the system which takes plaint text and key as input and encrypts data into ciphertext.

$Decrypt_\varepsilon$: Decryption algorithm for the system which takes cipher text and key as input and produces plain text as output.

$Evaluate_\varepsilon$: Evaluate algorithm takes public key pk, a circuit C from a permitted set of $C_\varepsilon$ of circuits, and a tuple of ciphertext$\Psi = (\psi_1, \ldots, \psi_t)$ and produces a ciphertext ψ as output.

The computational complexity of all these algorithms should be polynomial in security parameter λ and the size of C (in the case of $Evaluate_\varepsilon$).$\varepsilon$ is correct for circuits in $C_\varepsilon$ if for any key-pair (sk, pk) output by KeyGen(λ), any circuit $C \in C_\varepsilon$ , any plain texts $\pi_1 \ldots, \pi_t$ and any ciphertexts $\Psi = (\psi_1, \ldots, \psi_t)$ with$\psi_i \leftarrow$ Encrypt(pk, $\pi_i$), It can be written as

$\psi \leftarrow Evaluate_\varepsilon(pk, C, \Psi) \Rightarrow C(\pi_1 \ldots, \pi_t) = Decrypt_\varepsilon(sk, \psi)$

Only correctness does not eliminate trivial schemes. Therefore, we need ciphertext size and decryption time to be upper bound by a function of the security parameter λ independent of C.

1. Homomorphic encryption: ε is homomorphic for circuits in $C_\varepsilon$. If ε is correct for $C_\varepsilon$ and $Decrypt_\varepsilon$ can be expressed as a circuit $D_\varepsilon$ of size poly (λ).
2. Fully Homomorphic encryption: ε is fully homomorphic if it is homomorphic for all circuits.
3. Correctness: A Homomorphic encryption ε is correct for a family $C_\varepsilon$ of circuits if any pair(sk,pk) output by $KeyGen_\varepsilon(\lambda)$

any circuit $C \epsilon C_\varepsilon$ , any plain text $\pi_1 \ldots, \pi_t$ and any cipher text$\Psi = (\psi_1, \ldots, \psi_t)$ with $\psi_i \leftarrow$ Encrypt(pk, $\pi_i$), It can be written as

$\psi \leftarrow$Evaluate$_\varepsilon$ (pk,C,Ψ) ⇒ C($\pi_1 \ldots, \pi_t$) = Decrypt$_\varepsilon$(sk,ψ)

4. Compactness: A Homomorphic encryption ε is compact if there is a polynomial f so that, for every value of the security parameter λ, ε's decryption algorithm can be expressed as a circuit $D\varepsilon$ of size at most $f(\lambda)$.

5. Leveled Fully Homomorphic encryption: A family of schemes $\{\varepsilon^{(d)} : d \epsilon Z^+\}$ is leveled fully homomorphic if they all use the same decryption circuit, $\varepsilon^{(d)}$ is homomorphic for all circuits of depth at most $d$ that use some specific set of gates Γ), and the computational complexity of $\varepsilon^{(d)}{}'s$ algorithms is polynomial in λ, d, and the size of C(in the case of Evaluate$_\varepsilon$(d)).

## III. HOMOMORPHIC ENCRYPTION THROUGH LATTICES

The earlier homomorphisms in the encryption schemes were implied but never used for the purpose of utilizing these homomorphic properties for certain applications like delegating of computations. Also, the schemes were partially homomorphic having only limited properties. This section describes the schemes that are full homomorphic and are considered the early works that laid foundation for development of homomorphic cryptography as a separate branch of cryptography.

### *A.Gentry's blueprint scheme*

In 2009 Gentry [2] came up with the first fully homomorphic encryption scheme. It is based on lattice algebra. It contained primitives for encryption, decryption and methods to perform addition and multiplication that were homomorphic to the XOR and AND operations of plaintext bits. They also included the refresh operation to reduce the size of ciphertext whenever the noise in the ciphertext increased beyond certain level. The main concepts that were laid in the Gentry's blueprint were:

- Refresh method should be so formulated that it is a process being performed over the ciphertext and hence a homomorphic method to the actual decryption.
- Public key should be used for refreshing the ciphertext.
- Refresh means to encrypt the ciphertext under another key and hence is like recryption.
- The mathematical function chosen should be a trapdoor with homomorphic properties
- Scheme should be bootstrappable to have a refresh procedure.

This blueprint still guides the design of majority of the fully homomorphic schemes.

The drawbacks of this scheme were the impractical size of keys and costly encryption function.

### *B. Smart and VercauterenScheme*

Smart and Vercauteren [11] based their work on that of Gentry. It was a simple version of the lattice based scheme of Gentry. This scheme needs that the determinant of lattice should be a prime number. But it has advantage over Gentry's scheme like smaller key size and cipher text size and the value of expansion is also lower. It is fully homomorphic encryption scheme unlike Gentry which is somewhat homomorphic.

### *C. Gentry-Halevi Scheme*

A practical implementation of Gentry's work appeared in [12]. They suggested a modification or conclusion that the determinant of lattice should not be a prime number as was originally required in Gentry's work [2]. This condition was relaxed through certain modifications. For demonstration purposes the provided four parameter settings of sizes 512, 2048, 8192 and 32768. For the largest setting size 32768 the public key size is 2.3 GB that is very large. The author implemented this scheme on a high end work station after optimization the results are that encryption took 3 minutes and key generation took 2.2 hours.

### *D. Stehle and Steinfeld scheme*

This scheme was proposed in 2010 by Stehle and Steinfeld [13]. They provided a scheme which uses probabilistic decryption algorithm that can be implemented with an algebraic circuit of low multiplicative degree. This scheme improves the speed of encryption very much with $O(\lambda^{3.5})$complexity.

### *E. SIMD operations*

Smart and Vercauteren show [14] how to select parameters for Gentry's fully homomorphic encryption scheme implementation of 2011 [11] that uses Single Instruction Multiple Data (SIMD) operation, to construct a fully homomorphic scheme when performing re-encryption in parallel. They showed that their scheme is 2.4 times faster than the normal FHE schemes and ciphertext is reduced by 1/72. Thus, their scheme proves that efficiency can be increased through parallelism.

### *F. Gentry Halevi without squashing scheme*

This scheme was proposed in 2011, it uses a completely new technology to eliminate the squashing step of decryption of Gentry's SWHE scheme which is relied on bootstrapping. This scheme [14] was a combination of SWHE and MHE (Multiplicative Homomorphic Encryption). This scheme's main purpose is to express the decryption function of SWHE scheme as a depth of 3 arithmetic circuit of a particular form. Since the circuit is in particular form, it can be switched to MHE scheme without homomorphic evaluation of anything. The resultant can be converted to SWHE scheme by homomorphically evaluating decryption of MHE scheme. So, SWHE scheme should be able to evaluating MHE's decryption function only. Hence, it is avoiding the circulation that was important to squashing in the previous schemes of homomorphic encryption.

### *G. Gentry Halevi Smart scheme*

Gentry and Halevi [15] introduced a scheme in 2011 that modified the bootstrap process required for decrypting the ciphertexts. The homomorphic modular reduction of

*p modulo q* where p and q are integers which uses bit operation on binary representation of integers in bootstrap process was reduced to some extent by taking a modulus very close to power of 2. This scheme is fast and easier for implementation. And sometimes it encrypts the secret key as one ciphertext, so that the public key size can be reduced. This scheme can also be implemented with the previous SIMD homomorphic scheme model.

## IV. HOMOMORPHIC ENCRYPTION THROUGH APPROXIMATE GCD

Inspired by the difficult problem of factorization of large numbers and their GCD solution through Chinese Remainder Theorem, the security assumption of approximate GCD is adopted. Based on this security assumption, many fully homomorphic schemes using integer arithmetic and linear algebra were proposed.

### *A. Dijk, Gentry, Halevi and Vaikuntanathan*

This is a very different fully homomorphic scheme worked on integers proposed in 2010 [3]. It forms the basis of many further schemes that employ integer arithmetic for homomorphic trapdoors. The main feature of this scheme is that it is conceptually very simple. This scheme is also initiated as somewhat homomorphic scheme which performs a limited number of addition and multiplication operations on encrypted bits. But then it is converted into fully homomorphic scheme. In this all the operations (addition and multiplications) are done on integers rather than ideal lattice unlike Gentry's scheme. The main advantage of this scheme is that it uses integers in place of single bits for implementation, though the message space is yet a single bit.

The basic idea is to hide the message bit within an integer that is equivalent to the bit in some chosen Galois field. This is again expanded to become a number of another Galois field. The keys decide the Galois fields to be used. Decryption can then become a very simple operation of finding remainders when the ciphertext is divided by the public key. This forms blueprint of several FHE schemes.

The drawback is that message space is {0, 1} while the length of keys is several bytes. Such long keys are not of much practical use.

### *B. Coron, Mandal, Naccache, and Tibouchi scheme*

Coron et al [16] tried to reduce the size of keys used in the DGHV [3] scheme. Other modifications were also proposed to make it more practical. First of all it shows that the public key size of somewhat homomorphic encryption schemes can be reduced from $\tilde{O}(\lambda^{10})$ to $\tilde{O}(\lambda^{7})$. The concept was that they store only a smaller subset of public key and then generate the whole public key by multiplying all the elements of the subset. Approximate GCD is modified to get a stricter version .The semantic security of new scheme is proved under this assumption. The second improvement was to implement DGHV scheme with new variant and optimize it through Gentry and Halevi's idea [12]. They use refined analysis only from [13]. Their approach was to implement the known attacks and record their running time and derive for large parameters, so that the core parameters can be fixed to the desired security levels. They got the same results as Gentry's [12] especially for the four parameters defined by Gentry's [12]. The public key size was computed 800MB, decryption was immediate, encryption takes 3 minutes and re-encryption took 14 minutes. Therefore, it was concluded that homomorphic encryption can be applied with simpler techniques.

### *C. Chunsheng scheme*

Chunsheng[17] proposed a trial and error attack on FHE using integers by using lattice reduction algorithm. According to results we can directly get plain text from ciphertext by only using their public key(without secret key) for some parameter settings of FHE in DGHV[3] scheme. They did this by making a new lattice based on public key and convert cipher text from plain text by using LLL reduction algorithm. They proposed a scheme to avoid the above attack. In this scheme secret key is in the form of matrix rather than a large integer. Now to implement FHE we need to add secret key ciphertext with public key ciphertext. Sizes of secret key and public key are $O(\lambda^2)$ and $O(\lambda^3 \log \lambda)$ respectively.

### *D. Coron, Lepoint and Tibouchi scheme*

This scheme proposed in 2012 [4]. It takes DGHV scheme as base and provided batch capabilities to fully homomorphic encryption. Means that it manipulates a vector of plaintext bits homomorphically into a single ciphertext. It can perform arbitrary permutation on the plain text with the help of cipher text and public key only. And still maintains semantic security. It presents the approach of using features of LWE-based FHE scheme into Approximate GCD based scheme.

### *E. Cheon et al's scheme*

This scheme proposed by Cheon et al [18] in 2012, represents the new methods to implement a FHE scheme by combining the ideas of R. Rivest, L. Adleman, and M. Dertouzos[4] and M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan[10]. This scheme supports SIMD techniques and reduced the computation overhead although it has larger plain text.

## V. HOMOMORPHIC ENCRYPTION THROUGH RING LEARNING WITH ERRORS

A stronger assumption of security that consisted of a list of prime numbers and keys derived from them having no fast solution in approximate GCD is the ring learning with errors. Certain practical FHE schemes were designed under this assumption, with costly key generation and setup modules.

### *A. FHE-LWE scheme*

Schemes based on Gentry's schemes has a disadvantage of having high per gate evaluation time and large key sizes which does not allow them for practical usage of their FHE. Hence a research in new direction is needed. Brakerski and Vaikuntanathan [19] in 2011 used a new re-linearization technique and showed that FHE can be achieved through the

hardness of learning with errors (LWE) problem, which is way more standard than the previous techniques. This scheme uses a new element modulus reduction technique in place of squashing which reduces the decryption complexity as well as the size of the ciphertexts without any additional assumption. The authors also showed that this scheme can be used to make LWE-based single-server private information retrieval (PIR) protocol which is very efficient asymptotically.

*B. Brakerski, Gentry and Vaikuntanathan*

Brakerski, Gentry and Vaikuntanathan(BGV) [20] work on a scheme in 2011, which was based on FHE-LWE scheme [19]. Their gate per computation is approximately linear in security parameter. Hence it is linear efficient asymptotically. This scheme has a major difference with previous scheme that it iteratively uses modulus switching instead of doing it at one time only for shorter ciphertext size. The modulus size new this scheme is relatively large but noise level is constant. The major disadvantage of this scheme is that the homomorphic capability is also decreased due to larger modulus size.

*C. Lauter, Naehrig and Vaikuntanathan*

Lauter, Naehrig and Vaikuntanathan implemented FHE-LWE scheme [21] and published their observations and proposed many optimization which are specific for specific applications. Their popular contribution was that they showed that conversion between many message encoding in a ciphertext can be done and techniques to do that as well. The first technique is that integers are encoded in ciphertexts, which allows performing addition and multiplication operations on encoded integers efficiently. It allows to effectively computing mean, standard deviation and other private statistics. The second technique is to combine $n$ different encryptions of bits into one encryption string of $n$-bits. Encryption in bits is done since some homomorphic operations need bit-wise encryption like, comparison of integers and private information retrieval etc. In this method the result can be combined into a single encryption after performing these encryption operations.

*D. HELib*

IBM company has recently released a software package called HELib [22] in April 2013, it contains implementations of Homomorphic encryption (HE) techniques. They have uses different techniques combined. It is the implementation of BGV scheme [26], with many optimization to make homomorphic technique faster, used cipher packing technique of Smart-Vercauteren and optimizations of Gentry-Halevi-Smart scheme. This library is written in C++ and uses the NTL mathematical library (version 10.0.0 or higher). As of March 2015, it also supports multi-threading. At its present state, this library is mostly meant for researchers working on HE and its uses. Also currently it is fairly low-level, and is best thought of as "assembly language for HE". That is, it provides low-level routines (set, add, multiply, shift, etc.).

## VI. OTHER HE SCHEMES

*A.Domingo-Ferrer scheme*

Though earlier than Gentry's blueprint [2], this scheme by Domingo-Ferrer [1] did not receive much attention. It is a privacy homomorphism that provide both the addition and multiplication operations on ciphertexts. The public parameters required are a positive integer > 2 for security reasons and a very large integer that should have many small divisors. This scheme was proven to provide security against known ciphertext attacks but the condition was that the ciphertext space needs to be larger than the plaintext space. He also proposed some applications .But the scheme was not implemented practically because of large cipher text size.

*B. Ramaiaah and Vijayakumari*

Ramaiaah and Vijayakumari proposed a scheme in 2012 [5], this scheme was a very efficient fully homomorphic scheme that provide encryption over integers instead of bits. The security of this scheme was based on Approximate GCD. The main advantage of this scheme was that the public key size was very small, $\tilde{O}(n^4)$ as compared to Gentry's[2] and Coron et al[4] schemes. Overall Computation complexity was $\tilde{O}(n^8)$.

*C. Xiao et al*

Xiao et al proposed a scheme [7] in 2012, which was symmetric key homomorphic scheme which was non-circuit based. The scheme's main focus was to achieve efficiency hence it reduces security requirements. This scheme does not provide semantic security. But its security is as strict as large integer factorization problem. He also proposes that many user keys can be achieved from one master key to retrieve data from the server, instead of using only one master key. The data is encrypted using one master key. Then different keys are derived from that one master key and are assigned to each user. They also propose a protocol to ensure correct and secure communication using these user keys between server and user.

*D. Gahi et al*

Gahi et al [23] introduces FHE with circuits based first secure database system prototype in 2011, in which circuits are used for implement SQL queries over encrypted data. Data is stored and processed in the encrypted form without having knowledge of where is record actually stored in the database. But this technique was impractical since FHE produced impractically large cipher text so the time taken to process the SQL queries was also very large.

Gahi et al [24] in 2012 also proposed highly secure FHE based Location based service (LBS) which can process encrypted data to get location related data which can only be decrypted by the user who requested the data. It also provides security services related to data confidentiality and integrity. The proposed circuit uses search circuits. The executor (LBS server) receives the encrypted input and then performs search operation and retrieve the data matches the search query. The requestor can send the user location as well as service in encrypted form to the server and server search the data and send it to him without knowing the

actual content. The author also proposed an improved version of it.

*E. Brenner et al*

Brenner et al [25] produces a scheme in 2011, which can compute a secret program on third party server using FHE. His main focus was on data security on cloud. He also implemented his scheme as computer software which can solve problems of encrypted storage access with encrypted addresses and encrypted branching. They also provide the parameter so that an encrypted program and assembler can generate encrypted machine code.

*F. Wei and Reiter*

Wei and Reiter [26] proposes scheme in 2012, so that client authorized by the owner of data, can perform Deterministic finite Automaton (DFA) on encrypted data store on server (cloud). Their main focus was to outsource data and provide restricted access to data usage to a particular client (untrusted third party). This scheme provides the security to the data and DFA operation from pernicious server as well as malicious users.

*G. Damgard et al*

Damgard [27] proposes a scheme which was based on HE and does not use random oracles. This scheme provides an approach to achieve the compilation of class Σ-protocols into non-interactive zero-knowledge arguments. This scheme can be used in all discrete-log based Σ-protocols. By applying this scheme they observed a non-interactive threshold RSA and non-interactive zero knowledge for NP with high efficiency.

*H. Gauraha et al*

In 2013, Gauraha et al [28] proposes a FHE scheme which is very secure and provides confidentiality, duplicitous pliability and competence security services. This scheme uses robust algorithm for numerical stability, uses dense matrix and sparse matrix structure of problem to improve efficiency and uses stochastic chain process for encryption. Results showed that this scheme is practical.

*I. Gupta and Sharma*

A symmetric FHE scheme proposed in [6] and [29] that worked with matrices. They highlighted the need of symmetric FHE and situations where only symmetric schemes can provide the security. The message is hidden as eigenvalue of 4-by-4 square matrix such that there are other Eigen values possible. Probabilistic encryption is used. But decryption is able to retrieve the correct eigenvalue that corresponds to the message. The key is also a matrix. The homomorphic operations are matrix operations.

Also, a protocol is proposed as to how to use the FHE scheme for third party delegation of computation in cloud computing context. The proposal has a novel idea of using multiple keys. A message when encrypted using one key can be encrypted again using another key, by treating the previous ciphertext as a message. Yet, a single decryption using a master key can provide the plaintext. Also, partial decryption is possible when access to only certain users is to be given. For this keys at different levels are used. And only authorized users at a level can decrypt the ciphertext to proper plaintext.

*J. Aggarwal et al*

Aggarwal, Gupta and Sharma [8] in 2014 proposed a symmetric FHE scheme inspired from the DGHV blueprint [3]. They propose a homomorphic trapdoor using linear algebra with only modular arithmetic operations. The decryption procedure need not be bootstrappable in this case. Rather the design is just as in conventional cryptography where the decryption method inverts the process of encryption such that the functions are one way and knowledge of decryption does not imply knowledge of encryption. A refresh process involving another secret key that does not leak the actual key is also proposed in the work [8].

As an application the authors have demonstrated the use of symmetric FHE for private querying into a database stored at third party like a cloud and in encrypted form. The security of this scheme is based on hardness of factorization of integers.

*K. Kumar et al*

Kumar, Buksh and Sharma [30] propose how DGHV blueprint [3] can be modified and used for double encryption to secure files stored at third party like Cloud service provider. Their assumption is that files belong to an incremental text model which is followed by majority of confidential documents. Encryption is more similar to a homomorphic hash function giving a probabilistic encryption. A protocol is also proposed to detect if files have been tampered, and the tampering can be located up to single file level or up to a word of file level.

*L. Dasgupta and Pal*

Recently, Dasgupta and Pal [9] have proposed a somewhat homomorphic scheme which is made fully homomorphic using a refresh procedure. It extends the amount of message that can be encrypted at a time by encrypting messages as coefficients of a polynomial. After certain amount of homomorphic computations, large ciphertexts are refreshed for proper decryption. The hardness of the scheme is based on the difficulty of factorizing large integers. Also, it requires polynomial addition which is computationally cost effective. Experimental results are shown to support the claim but with very small values of parameters. Practically the parameters need to be large sized for desired level of security and may not have a practical feasible time for encryption and homomorphic operations.

## VII. CONCLUSION

Homomorphic encryption has not yet reached to the level of practical use even in large and advanced cloud systems. Though, IBM research lab has released a library of primitives to implement fully homomorphic encryption, it is being mainly used for research purpose. The main hurdles are long keys and high time complexity. Also, homomorphic operations required for database querying are not much developed. Operations like search, partial match, sorting etc. are not yet possible over encrypted texts. Standardization of

security notions is required. Equivalent homomorphic primitives for digital signatures, key-rings, group keys are still not in place. Public key FHE schemes have been more explored. Very few works exist for symmetric key FHE. The message space for which the prevailing schemes work is too small.

## REFERENCES


[1] J.I. Domingo-Ferrer, "A new privacy homomorphism and applications," Information Processing Letters 60.5 (1996): pp 277-282.

[2] C. Gentry, "Fully homomorphic encryption using ideal lattices", Proceedings of the 41st ACM Symposium on Theory of Computing ( STOC 2009), pp. 169-178, 2009.

[3] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. "Fully homomorphic encryption over the integers", *Proceedings of Eurocrypt-10, Lecture Notes in Computer Science*, vol. 6110, pp 24-43, 2010.

[4] J.S. Coron, T. Lepoint and M. Tibouchi, "Batch Fully Homomorphic Encryption over the Integers", Cryptology ePrint Archive: Report 2013/036, 2013.

[5] Y.G. Ramaiah and G. VijayaKumari, "Efficient public key generation for homomorphic encryption over the integers," Advances in Communication, Network, and Computing. Springer Berlin Heidelberg, 2012 pp 262-268

[6] C.P. Gupta and I. Sharma, "A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds," Network of the Future (NOF), 2013 Fourth International Conference on the , vol., no., pp.1,4, 23-25 Oct. 2013 doi: 10.1109/NOF.2013.6724526.

[7] L. Xiao, O. Bastani and I-L. Yen, "An Efficient Homomorphic Encryption Protocol for Multi-User Systems," IACR Cryptology ePrint Archive 2012 (2012): 193.

[8] N. Aggarwal, C.P. Gupta and I. Sharma, "Fully Homomorphic Symmetric Scheme without Bootstrapping", Proceedings of International Conference on Cloud Computing and Internet of Things (CCIOT), 14-17, 2014.

[9] S. Dasgupta and S.K. Pal, "Design of a polynomial ring based symmetric homomorphic encryption scheme", Perspectives in Science (2016) 8, 692—695.

[10] J. Benaloh, "Dense Probabilistic Encryption (PS)", Workshop on Selected Areas of Cryptography. pp. 120–128, 1994.

[11] N. P. Smart and F. Vercauteren. "Fully homomorphic encryption with relatively small key and ciphertext sizes", Public Key Cryptography – PKC 2010, Berlin, Heidelberg, New York, 2010. [Lecture Notes in Computer Science, vol 6056, Springer Verlagpp 420–443]

[12] C. Gentry and S. Halevi. "Implementing Gentry's fully homomorphic encryption scheme." EURO-CRYPT 2011, LNCS, Springer, K. Paterson (Ed.),2011.

[13] D. Stehle and R. Steinfeld. "Faster fully homomorphic encryption." Cryptology ePrint Archive Report 2010/299.

[14] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations", IACR Cryptology ePrint Archive, Report 2011/133.

[15] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits", Cryptology ePrint Archive, Report 2011/279

[16] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. "Fully homomorphic encryption over the integers with shorter public-keys", Advances in Cryptology - Proc. CRYPTO 2011, vol. 6841 of Lecture Notes in Computer Science. Springer, 2011.

[17] G. Chunsheng, "Attack on Fully Homomorphic Encryption over the Integers", 2012. Available at https://arxiv.org/abs/1202.3321

[18] J-H. Cheon, J. Kim, M-S. Lee and A. Yun, "CRT-based fully homomorphic encryption over the integers" *in Information Sciences*, Volume 310, Pages 149–162, 2015.

[19] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from(standard) LWE," in Foundations of Computer Science, 2011. Also available at Cryptology ePrint Archive, Report 2011/344.

[20] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. "Fully homomorphic encryption without bootstrapping", Cryptology ePrint Archive, Report 2011/277.

[21] K.Lauter, M.Naehrig and V.Vaikunthnathan, "Can homomorphic encryption be practical?",Proc of 3rd ACM workshop on Cloud Computing Security Workshop , pp 113-124, 2011.

[22] https://github.com/shaih/HElib

[23] Y. Gahi, M. Guennoun and K. El-Khatib, ""A Secure Database System using Homomorphic Encryption Schemes" *in The Third International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA 2011*, 2011.

[24] Y. Gahi, M. Guennoun, Z. Guennoun and K. El-Khatib, "Privacy Preserving Scheme for Location-Based Services" in Journal of Information Security, 3, 105-112, 2012

[25] M. Brenner, J. Wiebelitz, G.Voigt and M. Smith, "Secret Program Execution in the Cloud Applying Homomorphic Encryption" *in Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011)*, Korea, 2011

[26] L. Wei and M. K. Reiter, "Third-Party Private DFA Evaluation on Encrypted Files in the Cloud" *in Proceedings of the 17th European Symposium on Research in Computer Security, Pisa, Italy, Volume 7459 of the series Lecture Notes in Computer Science*,pp 523-540, 2012

[27] I. Damgard, N. Fazio and A. Nicolosi, "Non-interactive Zero-Knowledge from Homomorphic Encryption" *in Proceedings of the Third Theory of Cryptography Conference, TCC 2006, New York, Volume 3876 of the series Lecture Notes in Computer Science*pp 41-59, 2006.

[28] N. Gauraha, D. Mishra and P. Trivedi, "Data Security in Distributed System using Fully Homomorphic Encryption and Linear" in International Conference on Communication Systems and Network Technologies (CSNT), pgs- 423-425, doi- 10.1109/CSNT.2013.94, 2013.

[29] I. Sharma, C.P. Gupta, "Making Data in Cloud Secure and Usable: Fully Homomorphic Encryption with Symmetric keys" in International Journal of Communication Networks and Distributed Systems, Inderscience. 2015 Vol. 14 No. 4 pp 379-399. DOI: 10.1504/IJCNDS.2015.069673, 2015.

[30] V. Kumar , B. Buksh and I. Sharma, "Double Fully Homomorphic Encryption for Tamper Detection in Incremental Documents". In: Satapathy S., Das S. (eds) Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2. Smart Innovation, Systems and Technologies, vol 51. Springer, Cham, 2015.